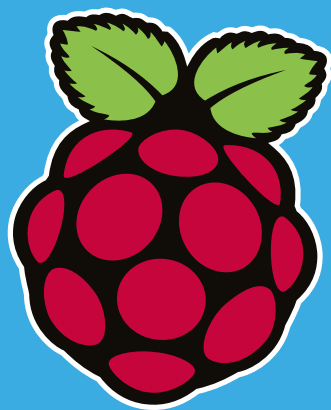
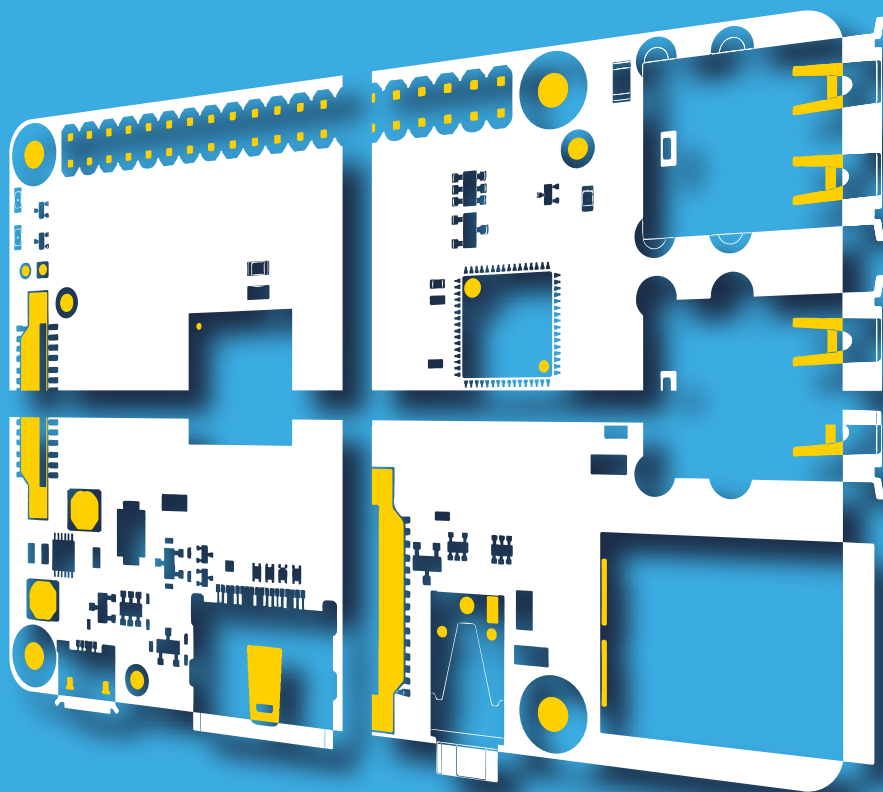
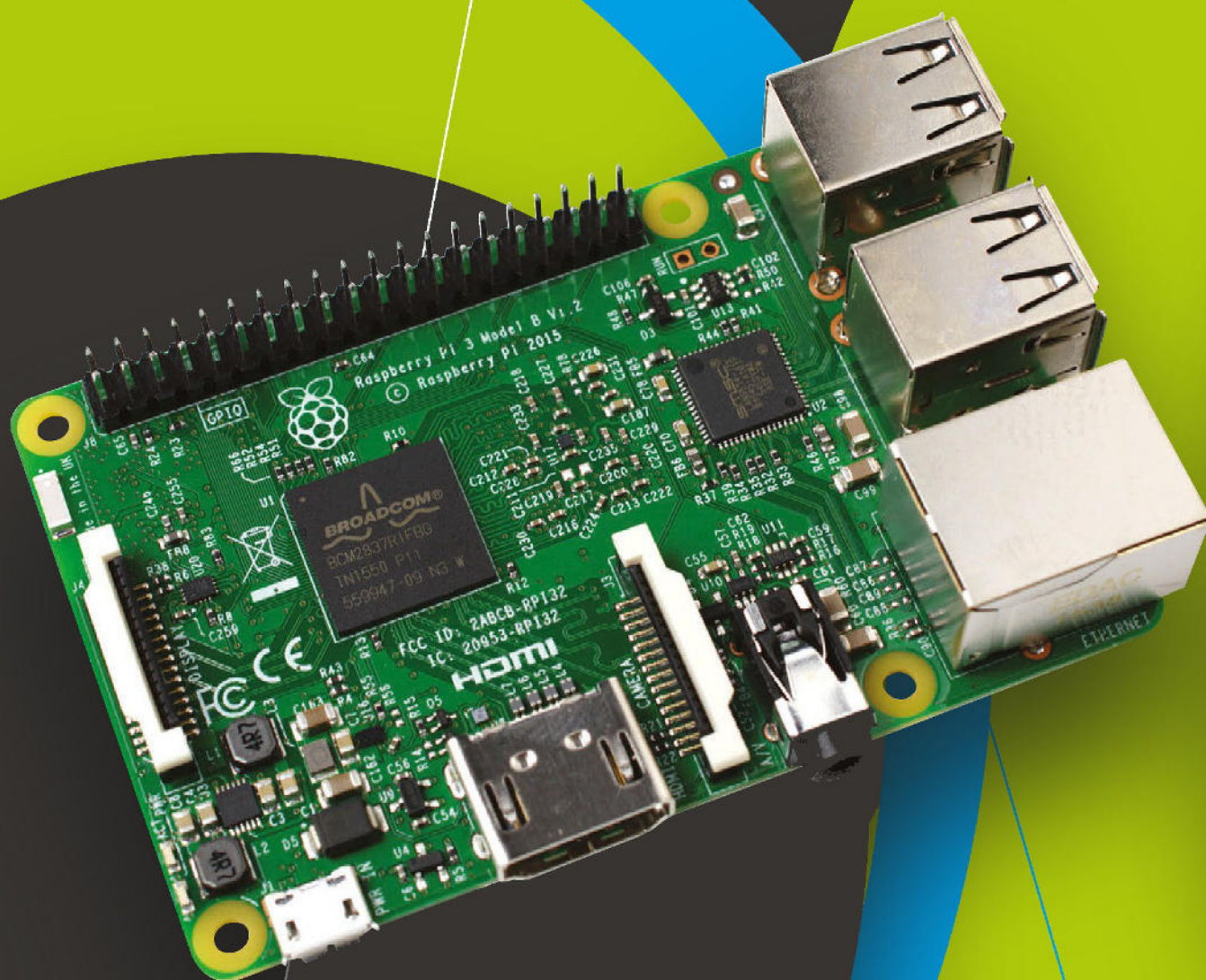
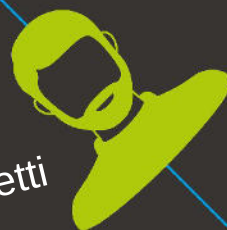
LA TUA RIVISTA UFFICIALE DI RASPBERRY PI

La rivista ufficiale Raspberry Pi
in italiano, da RaspberryItaly.com

Numero 48 Agosto 2016 www.raspberryitaly.com

WINDOWS 10 IoT CORE SU RASPBERRY PI

Crea dei dispositivi dell'Internet delle cose con la nuova e entusiasmante realizzazione di Microsoft

L' UNICA RIVISTA RASPBERRY PI SCRITTA DAI LETTORI, PER I LETTORI

“Ottieni di più dal tuo Raspberry Pi!”
Impara ad usare “IoT Core!”

« Scopri Fantastici progetti »

WINDOWS 10 IoT CORE

Il Raspberry Pi 3 apre a Windows 10 IoT Core per più ampi utilizzi – ecco come partire...

In occasione del lancio di Raspberry Pi 3 a Londra , il team di Microsoft era presente per sfoggiare il nuovo progetto che avevano creato per promuovere l'uso del Pi e di Windows 10. Grazie ad esso, hanno dimostrato come realizzare un ciclo di regolazione per mantenere una ruota in rotazione grossomodo alla stessa velocità, anche con resistenza o sforzo variabili, può non sembrare sorprendente, ma per permetterne la continua rotazione, è stato necessario molto lavoro "dietro".

Questa è la forza di Windows 10 IoT Core sul Raspberry Pi 3, che consente più potenza per realizzare dei progetti unici che si connettono all'Internet delle Cose. Con un nuovo kit in arrivo per aiutare le persone a iniziare con Windows 10 su Raspberry Pi 3, abbiamo pensato di portarci avanti, per darti un vantaggio in partenza, in modo che tu possa cominciare a fare progetti più incredibili e vari sul tuo Raspberry Pi.

RASPBERRY PI E WINDOWS

Ecco come Microsoft è pronta a sostenere i futuri maker con Windows 10 IoT Core

Microsoft era presente al lancio del Raspberry Pi 3 per mostrare il suo nuovo progetto della ruota.
Scopri l' interno funzionamento, qui : magpi.cc/2a8rak9

CONTENUTO DEL BOX

Il nuovo kit ha una grande ricchezza di componenti per iniziare fare alcune cose incredibili con Windows 10 IoT Core:

- > Display da 5 pollici
- > Grove shield (per collegarvi i componenti)
- > Sensore a ultrasuoni
- > Sensore di suono
- > Sensore di temperatura e umidità
- > Sensore di luce
- > Sensore rotazione angolare
- > Retroilluminazione per LCD RGB
- > Barra LED
- > Relè
- > Pulsanti
- > Ronzatore
- > Cavo HDMI

Microsoft lancia un nuovo kit per Raspberry Pi che consente di realizzare alcuni progetti incredibili con Windows 10 IoT Core. Abbiamo raggiunto Asobo Mongwa e Dan Rosenstein della squadra Microsoft che sta dietro al kit, per scoprire come sono giunti alla sua realizzazione.

The MagPi: Come si è giunti ad avere Windows IoT Core su Raspberry Pi?

Dan Rosenstein: Raspberry Pi stava partendo sul mercato e nello stesso tempo stavamo lavorando a Windows 10 IoT Core.
Fin dall'inizio, la nostra convinzione è stata che i maker sono il motore creativo e il motivo di svolta per l'IoT, e che il movimento maker stava per essere il catalizzatore per questo cambiamento. Il nostro obiettivo progettuale per Windows 10 IoT Core era di creare un modello di programmazione semplice e accessibile ai maker, e che funzionasse con il grande insieme di hardware democratico già presente nell'ecosistema dei maker. Il nostro obiettivo era quello di raggiungere i maker dove essi si trovavano. La Fondazione Raspberry Pi ha fornito il perfetto mix per lanciare Windows 10 IoT Core : aveva l'hardware, la quota di mercato, le collaborazioni e la comunità di cui volevamo essere parte.

Il nuovo Grove kit di Microsoft è un pacchetto speciale che consente di fate la maggior parte dei progetti per Windows 10 IoT Core su Raspberry Pi

INCONTRA IL TEAM

Dan Rosenstein (a sinistra) è un Principal Technical Program Manager e Asobo Mongwa (a destra) è un Program Manager; entrambi lavorano al Gruppo Sistemi Operativi di Microsoft nella squadra IoT Maker Program Management. Qui sono alla National Maker Faife a Washington, DC in Giugno: uno dei tanti eventi Maker a cui la squadra ha partecipato al fine di mostrare la potenza di Windows 10 IoT Core su Raspberry Pi alla comunità dei maker.

TM: Quale è stata la reazione della community?

Asobo Mongwa: Siamo lieti di vedere l'entusiasmo nella comunità maker, come acquisisca sempre più familiarità con l'investimento che Microsoft sta facendo nell'internet delle cose, e i continui miglioramenti che stiamo apportando a Windows 10 sul Raspberry Pi. Abbiamo sentito questa eccitazione in prima persona durante la recenti fiere Maker nella Bay Area e a Washington DC. Non vediamo l'ora di vedere come i maker proseguiranno a creare nuove innovazioni che utilizzano Windows 10 IoT Core e il Pi 2 e 3, soprattutto ora che abbiamo condiviso informazioni sul nostro Market per percorsi di mercato: soluzioni che abbiamo messo in atto per consentire ai maker di accelerare la commercializzazione della loro idea sul nostro Market.

TM: Quale è stata l'idea dietro al nuovo kit?

EQ l Il Groove kit in arrivo da Microsoft, è una collaborazione con Seeed Studio. Seeed Studio ha dimostrato grande interesse per gli investimenti di Microsoft nell'IoT e nell'universo maker, e ci ha avvicinato per un impegno di co-engineering sul kit. Lo vediamo come un indicatore del fatto che coloro i quali influenzano il settore riconoscono l'impatto nel lavoro che stiamo facendo, e vogliono essere partner con noi per aiutare l'ecosistema a rilasciare la prossima generazione di soluzioni sul Raspberry Pi con Windows 10 IoT Core. Il Grove kit serve a rendere le cose più facili, ed è progettato per gli sviluppatori hardware e software a tutti i livelli di competenza tecnica. Cosa personalmente mi piace,a questo proposito, è il fatto che toglie complessità come la saldatura, cavi jumper e basette, e consente all'utente di concentrarsi sulla propria creatività. E ' modulare, quindi puoi utilizzare i componenti inclusi in questo Kit così come altri componenti aggiuntivi Grove.

TM: Quanto può fare in più un Pi 3 rispetto ai precedenti?

HVl Ci sono state un paio di fondamentali modifiche con il Pi 3, tutte sono state abilitate su Windows 10 IoT Core: clock più veloce, controller SD migliorato, WiFi e Bluetooth integrati. Grazie al clock più veloce, le prestazioni de l GPIO con il nostro Lightning driver si sono assestate a un livello decisamente più elevato.

TM: Quanto è importante il rapporto tra Windows 10 e il Raspberry Pi, per Microsoft?

EQ l Raspberry Pi ha avuto un impatto significativo nel campo dell'istruzione e nel movimento maker. Siamo entusiasti di vedere l'aumento di interesse per il mercato industriale e per la progettazione di prodotti finali . Noi collaborato con Element 14 per offrire il Pi 3 tra le piattaforme hardware supportate da Microsoft per la commercializzazione con Windows 10 IoT Core. Il Raspberry Pi è stato mostrato in alcune delle conferenze importanti di Microsoft, come //build, e la Windows Hardware Engineering Community conference in Cina. Questi sono eventi in cui condividiamo gli investimenti di ingegneria attuali e futuri di Microsoft con i partner hardware e software..

DAVIDE LONGO

Davide è uno sviluppatore software .NET presso il dipartimento informatico di una compagnia aerea. Vive nella bellissima isola della Sardegna, è laureato in fisica ed è appassionato di nuove tecnologie.

magpi.cc/2a30GRI

"Atmosfera per l'era moderna!"

CROMOTERAPIA

Questo progetto futuristico utilizza Windows 10 IoT su un Raspberry Pi per monitorare la tua felicità. Inoltre, regola la luminosità della stanza a seconda del tuo umore.

Cosa Serve

- Arduino Uno arduino.cc
- Striscia LED RGB adafruit.com/product/306
- Bluetooth HC-05 amazon.it/dp/B01G9KSAF6
- Breadboard amazon.it/dp/B00PQC72ZS
- Webcam

"Microsoft ha una nuova servizio che sfrutta il riconoscimento facciale" spiega lo sviluppatore .NET Davide Longo. "Questa funzionalità di sicurezza elabora e fornisce informazioni sotto forma di punteggio, valutando i livelli di felicità, rabbia, disprezzo, paura, disgusto, sorpresa e tristezza.

Questo servizio, parte dei Servizi Cognitivi di Microsoft chiamati 'Emotion API', ha ispirato Davide nella realizzazione di questo straordinario progetto per Windows 10 IoT. Cromoterapia controlla le espressioni facciali dell'utente e regola l'illuminazione ambientale di conseguenza.

Il progetto prevede un Raspberry Pi che abbia Windows IoT 10 Core al suo interno. Il Raspberry Pi è collegato ad una webcam, che invia le immagini del viso dell'utente alle API Emotion.

Queste ultime inviano a loro volta un messaggio, tramite Bluetooth, al sistema di illuminazione domestica, simulato utilizzando un Arduino collegato ad una striscia di LED RGB.

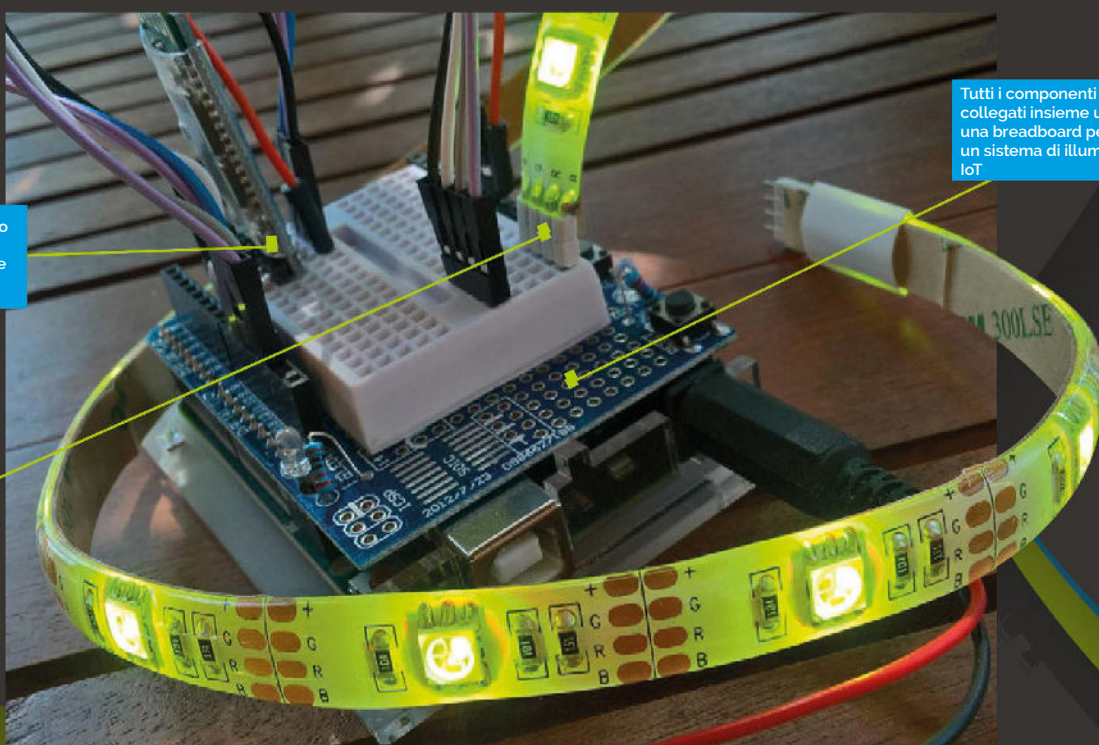
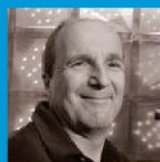
"Ogni volta che inizio un nuovo progetto, cerco di creare qualcosa di nuovo e di innovativo che utilizzi le tecnologie più recenti," afferma Davide. "Immediatamente ho pensato all'importanza della luce in un ambiente lavorativo."

"Nella cromoterapia, ci sono semplici regole per quanto riguarda i colori che si riflettono nell'esperienza quotidiana," spiega, "il blu è un colore calmante e rinfrescante; il verde simboleggia equilibrio, pace, e rinnovamento."

Tutto il codice è scritto in C# ed è disponibile su GitHub (**magpi.cc/2a31fLp**). "Condividere idee è la culla dell'innovazione futura," afferma Davide. "Nonostante quello che si possa pensare, tutti i programmi Microsoft e le funzioni Microsoft Azure qui utilizzate, sono libere e gratuite."

Il Raspberry Pi è collegato al sistema Cromoterapia tramite questo adattatore Bluetooth HC-05

Tutti i componenti sono collegati insieme utilizzando una breadboard per simulare un sistema di illuminazione IoT

Una striscia di LED RGB viene utilizzata per simulare il sistema di illuminazione

PHILIPPE LIBIOULLE

Originario del Belgio, Philippe è un manager del gruppo R&D che lavora in Canada. Ha creato una molteplicità di progetti di automazione domestica, e sta attualmente esaminando la tecnologia Thread utilizzando prodotti di Silicon Labs.
hackster.io/ThereIsNoTry

AGENTE DI SICUREZZA DOMESTICA PERSONALE

Fuoco e acqua sono un rischio per qualsiasi abitazione, ma un maker utilizza il suo Raspberry Pi con Windows 10 IoT Core per monitorare la sua casa e ricevere avvisi istantanei per qualsiasi problema.

Cosa Serve

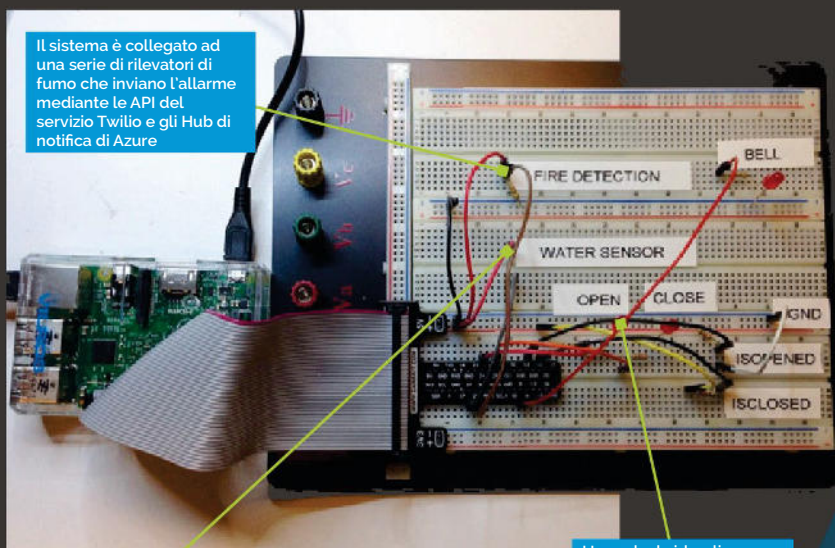
- Valvola acqua amazon.it/dp/B00XQMX2UO
- Sensore acqua magpi.cc/2a3v1io
- Allarme incendio magpi.cc/2a3ukpg
- Relè SSR magpi.cc/2a3uZXY
- Modulo relè rilevatore fumo magpi.cc/2a3v79J
- relè doppio e singolo 12V SPDT amazon.it/dp/B01FA1JR5C

Il Canada è un posto incantevole dove vivere, ma è allo stesso tempo un ambiente difficile. In inverno, la temperatura può scendere fino a -30°C, il che significa che il riscaldamento domestico deve rimanere acceso e i tubi possono rompersi per il gelo.

"Stavo parlando con un amico," afferma Philippe Libioulle. "Egli ha dovuto abitare lontano da casa per diversi mesi, solo perché una lavastoviglie non funzionava più. Durante una chiacchierata con un altro amico, mi sono reso conto che anche le perdite d'acqua costituiscono un grande problema. Il mio amico ha dovuto cambiare tutti i mobili del suo scantinato, a causa della rottura di un tubo mentre era a lavoro.

"I rilevatori di fumo sono obbligatori, stando alle norme, ma se non vi è nessuno in casa per intervenire, la segnalazione non viene inoltrata." Così Philippe si mise all'opera elaborando questo sistema di sicurezza domestica dell' "Internet delle Cose" (IoT).

Un Raspberry Pi esegue il sistema operativo Windows 10 IoT Core, e monitora gli allarmi antincendio e antifumo e le perdite d'acqua. Se venissero rilevati dei guasti, il sistema cloud attiverebbe le azioni necessarie. Queste includono l'accensione dell'allarme antincendio, o la chiusura del principale ingresso d'acqua all'interno della casa. Il sistema invia anche un messaggio al cellulare di Philippe, o ai suoi vicini di casa. "Se il sistema fosse abbastanza stabile, prenderei in considerazione anche la segnalazione al numero di emergenza 911," afferma Philippe.

Il sistema è collegato ad una serie di rilevatori di fumo che inviano l'allarme mediante le API del servizio Twilio e gli Hub di notifica di Azure

Un sensore dell'acqua al livello del suolo viene utilizzato per rilevare qualsiasi perdita in casa.

Una valvola idraulica elettronica è collegata al sistema. Se viene rilevata una perdita, la valvola disattiva il flusso d'acqua all'interno dell'abitazione.

> "Mantieni sicura la tua casa!"

"Ricevo avvisi tramite SMS, utilizzando le API del servizio Twilio (**twilio.com**). Utilizzo anche gli Hub di notifica di Azure (**azure.microsoft.com**) per inviare avvisi ai portatili, smartphone, e tablet.

"Sono ispirato dalle sfide concrete, espresse dalla gente reale," aggiunge Philippe. "Ho una nuova versione di questo progetto, più modulare e con un aspetto più definito."

JENS MARCHEWKA

Sviluppatore software C# di Dortmund, Jens ha molta esperienza con i sistemi operativi Windows, quindi è stato la scelta più ovvia per questo progetto.
magpi.cc/29ysoSK

SPECCHIO MAGICO

Specchio, specchio delle mie brame, mostrami le ultime novità e tutto il resto! Questo specchio, grazie al Pi, offre molto di più della tua immagine riflessa.

Cosa Serve

- GG Mirastar specchio unidirezionale magpi.cc/29FvoCo
- LCD monitor
- Legno per fare la cornice
- Cavi e connettori vari

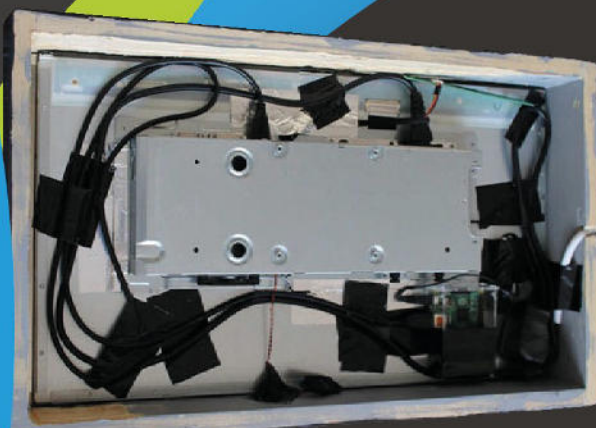
Nel tentativo di acquistare un vetro a specchio unidirezionale per il suo ultimo progetto, Jens Marchewka ha ricevuto alcune strane reazioni da parte dei fornitori che aveva contattato, con domande del tipo"Cosa vuole fare con questo specchio? No, non lo voglio sapere, arrivederci."

Jens aveva bisogno di quello specchio per creare uno 'specchio magico' che visualizza informazioni su uno schermo LCD dietro il pannello di vetro riflettente. Ispirato dal progetto precedente di Michael Teeuw (**magpi.cc/29y06Yk**), Jens voleva uno specchio che fosse in grado di mostrare degli articoli dalle sue fonti di notizie preferite, oltre ad informazioni sulle condizioni atmosferiche.

Non volendo inserire degli orrendi pulsanti frontali, Jens ha deciso, invece, di utilizzare comandi vocali composti da una sola parola per controllare il display dello specchio. A tale scopo, il sistema di riconoscimento vocale incorporato in Windows si è rivelato utile, nonostante Jens abbia riscontrato alcune difficoltà iniziali. "Stavo utilizzando il sistema di riconoscimento senza un set di regole grammaticali e ciò non ha funzionato correttamente... il sistema [ha continuato] a riconoscere le parole [che] nessuno

Reagendo ai comandi vocali, lo specchio visualizza informazioni utili come ad esempio gli eventi del calendario

Uno specchio a senso unico consente al monitor LCD posto dietro lo specchio di essere visto

Una cornice di legno su misura viene utilizzata per racchiudere lo specchio e le apparecchiature

L'interno dello specchio interamente istallato; in basso a destra è possibile vedere il Pi

stava pronunciando ed è stato anche attivato dalla radio. Questo è facile da evitare se si utilizza un set di regole grammaticali."

Per l'intero processo di costruzione gli ci sono volute solamente dieci ore, inclusa la costruzione del case di legno per incastrare lo specchio e incorniciare il monitor LCD, un Raspberry Pi 2, e i cavi di alimentazione nella parte posteriore dell'involucro in legno. Per cercare di tenere saldamente il tutto nella posizione desiderata, Jens ha aggiunto un profilo nella parte anteriore della cornice.

> "**Il futuro è qui** con internet nel tuo specchio!"

Ha poi dipinto il case di nero: "Mentre costruisci la cornice, dovresti ricordarti che la vernice occuperà spazio, quindi devi tenere ciò in mente e non costruirla troppo stretta."

Jens ha utilizzato un monitor LCD Iiyama da 24 pollici per il suo progetto, ma ciascun modello o dimensione potrebbe essere utilizzato, purché abbia un ottimo livello del nero, per il raggiungimento di un buon effetto specchio. Jens inoltre osserva che il monitor deve essere smontato dal suo case per posizionare il pannello LCD il più vicino possibile allo specchio. Mentre il suo schermo LCD è perfettamente e costantemente funzionante, Jens sta lavorando su una configurazione che programmi la sua alimentazione tramite il canale di comunicazione CEC.

Insieme alle notizie e al meteo, lo specchio magico di Jens mostra eventi del calendario – incluse le partite di calcio della sua squadra preferita, lo Schalke 04 –tramite il calendario web iCal. "Puoi impostare più di una sorgente iCal e assegnare un colore a ciascuna... ciò consente di avere molteplici eventi evidenziati in colori differenti." Jens ha aggiunto allo specchio anche una web radio con funzione di ripetizione, e ha intenzione di utilizzare il riconoscimento facciale per visualizzare eventi personalizzati e notizie per lui e sua moglie.

Gli eventi del calendario mostrati in alto a sinistra, e le informazioni meteo in alto a destra. Di' 'notizie' per vedere gli ultimi titoli

LA MAGIA DI **WINDOWS 10 IOT**

Design dell'interfaccia utente

Una delle ragioni principali per cui Jens ha scelto di utilizzare Windows 10 IoT Core per il suo progetto, insieme alla sua esperienza come sviluppatore Windows, è quanto sia facile elaborare interfacce utenti che sembrino perfette.

Riconoscimento vocale

Il sistema di riconoscimento vocale di Windows è utile per lo specchio magico. Jens ha semplicemente creato una funzione di Riconoscimento Vocale con una lista di comandi. Lo specchio esegue una sessione continua di riconoscimento vocale.

Syndication client

Le notizie dettagliate provengono dalle feed RSS, utilizzando la classe SyndicationClient sulla base dei modelli di Windows. "è una classe abbastanza soddisfacente per analizzare i feed RSS, " ci comunica Jens.

PRIMI PASSI CON WINDOWS 10 IOT CORE

Prepara il tuo Raspberry Pi 3 per lavorare con Windows 10 IoT Core

Prima di poter iniziare a utilizzare Windows 10 IoT Core, è necessario installarlo sul Raspberry Pi. Il metodo consigliato per farlo è attraverso la IoT Dashboard su un sistema Windows 10; questo ti consentirà di programmare poi facilmente il RaspberryPi con Windows 10, utilizzando Visual Studio. Iniziamo!

Nella Dashboard IoT, potrai anche trovare dei progetti di esempio e presto sarai in grado di connetterti al servizio cloud Azure IoT.

Segui le istruzioni sullo schermo, nella procedura guidata.

>PASSO-01
Preparare il PC

Per cominciare, assicurarti che il PC o il portatile che userai sia aggiornato all'ultima versione di Windows 10, o almeno alla versione 10.0.10240. Puoi controllarlo facendo clic sul pulsante Start e digitando **winver**. Una volta effettuato il controllo dell'aggiornamento, è necessario scaricare e quindi installare, la Dashboard IoT dal link qui : **magpi.cc/2ah13dt**

>PASSO-02
Recuperare l'immagine corretta

Per il Raspberry Pi 3, è necessario registrarsi al Windows Insider Program (**Insider.windows.com**). Successivamente dovrai trovare e scaricare l'ultima versione di Windows 10 IoT Core Insider Preview da qui (Scegli il numero più alto):
magpi.cc/29Ugtoz

Portati ora nella cartella di download e fai un doppio clic sul file ISO per entrare in esso, quindi fare clic sul programma di installazione all'interno, per far partire impostare correttamente il tuo PC per scrivere la scheda SD.

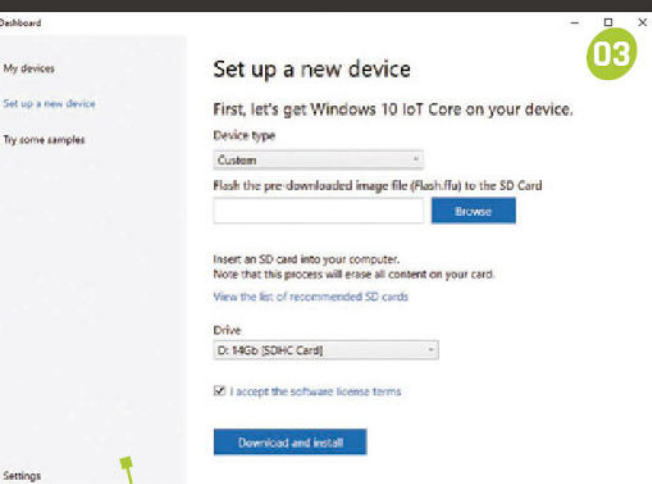
>PASSO-03
Scrittura sul disco

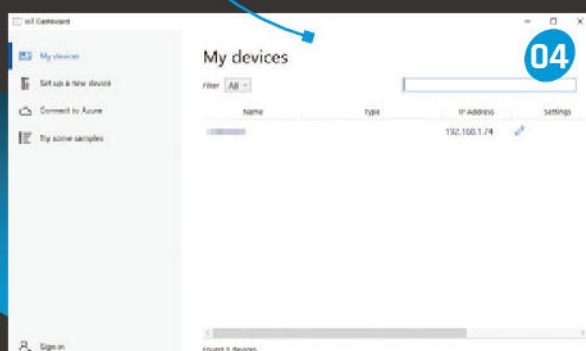
Una volta che il programma di installazione è terminato, lancia la Dashboard IoT che abbiamo scaricato al punto 1. Da qui siamo in grado di scrivere sulla scheda SD. Primo, fare clic su 'Configura un nuovo dispositivo'. Selezionare **Personalizzato** dal menù 'Tipo di dispositivo' e fare poi clic su **Sfoglia** per indicare il file immagine. Dovrebbe essere in:

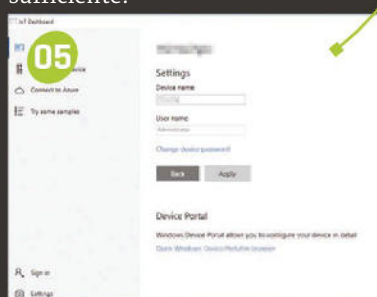
```
C:\Program Files (x86)\
Microsoft IoT\FFU\
```

WINDOWS 10 SUL RASPBERRY PI 2

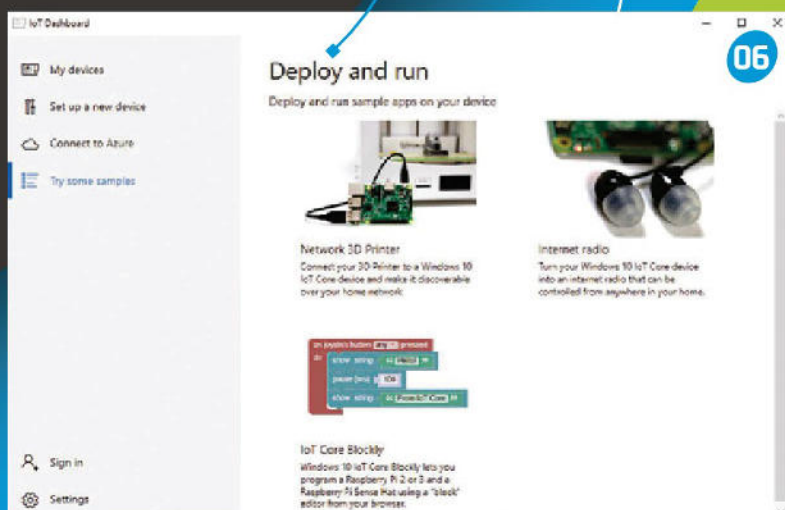
È possibile installare Windows 10 IoT Core in maniera molto similare anche sul Raspberry Pi 2; infatti, IoT Core è già disponibile nel NOOBS. Effettuare un upgrade a Raspberry Pi 3, però, è il miglior modo per fare uso di Windows 10 su Pi, per via della potenza di calcolo maggiore.

Lì, ci sarà una cartella Raspberry Pi, che contiene il file **Flash.ffu**. Selezionalo e poi fai click su Installa / Scarica e installa.

>PASSO-04
Networking

Per i nostri tutorial, e anche in generale, è necessario essere in grado di connettersi al Raspberry Pi tramite rete, per programmarlo facilmente e rapidamente. Si consiglia di collegarlo via Ethernet inizialmente, se possibile. Vai su 'I miei dispositivi' e dovresti trovare il Raspberry Pi, così da poterlo configurare, una volta acceso; questo include la creazione di una configurazione per l'accesso alla rete wireless per un utilizzo futuro.

>PASSO-05
Configurare la scheda

Una volta identificato il tuo Pi, è possibile configurarlo facendo clic sull'icona a forma di matita accanto ad esso. Da qui puoi lanciare Windows Device Portal, che ha alcune impostazioni di base che puoi modificare. Come la creazione di una rete wireless se ne hai bisogno, ed è anche una buona idea impostare uno Username e modificare la password predefinita. Quest'ultima è **p@ssw0rd**, è bene saperla, prima di cambiarla. Ha un livello di complessità appena sufficiente!

>PASSO-06
Start making

Sei pronto per andare avanti! Forse hai notato che esiste la possibilità di testare gli esempi nella colonna laterale, quindi se vuoi provare un progetto di base presente della lista, lo puoi fare. E' sufficiente selezionare il Raspberry Pi dal menu a tendina, e sei a cavallo.

Per programmarlo in futuro, lo connetteremo a Visual Studio; di questo parleremo nel nostro primo tutorial, oltre la pagina.

HAI UNA DIFFERENTE CONFIGURAZIONE?

Segui la guida personalizzabile Primi Passi qui: **magpi.cc/2abqzAW**

PHIL KING

Quando non fa sub-editing a The MagPi o scrive articoli, Phil ama lavorare su progetti per Pi, tra cui il suo robot a 2 ruote.
@philking68

LED ATTIVATO DA PULSANTE

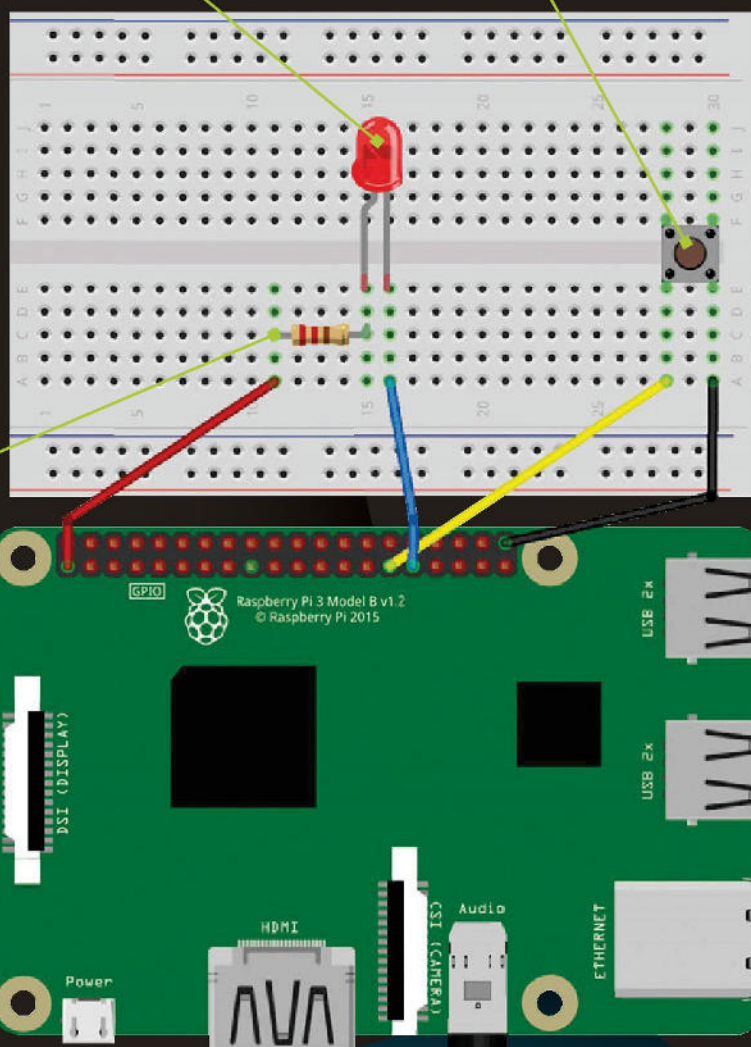
Crea un semplice progetto di physical computing utilizzando un LED e un pulsante

Cosa Serve

- 1× breadboard
- 1×pulsante
- 1× LED
- 1× resistenza da 333Ω
- 4× cavallotti maschio - femmina

Il LED è collegato al pin 3V3 tramite la resistenza e al GPIO 6

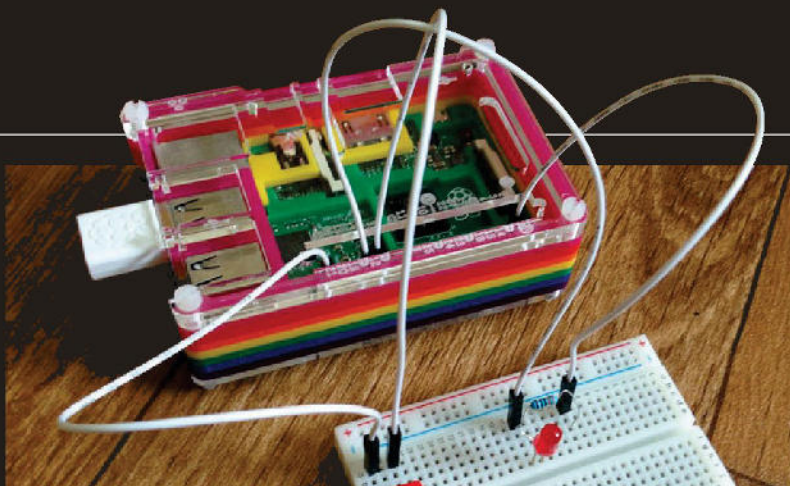
Può essere utilizzato qualsiasi pulsante; è collegato a GND e GPIO 5

Una resistenza da 330Ω serve a evitare una corrente eccessiva sul LED

Hai scaricato Windows 10 IoT Core, lo hai installato su una scheda SD, e hai impostato il tuo Raspberry Pi come un dispositivo di IoT Core? È il momento di iniziare a scrivere del codice, utilizzando Visual Studio 2015 su di un computer Windows 10 (PC). Cominciamo con un semplice progetto che utilizza un pulsante per attivare o disattivare un LED...

Connetti il tuo Pi

Prima di cominciare, hai bisogno di conoscere l'indirizzo IP del Raspberry Pi , mostrato ne 'I Miei Dispositivi' nella Dashboard di Windows 10 IoT Core. Avrai anche bisogno di Visual Studio 2015 Community installato, che puoi trovare qui: **magpi.cc/2adz7Hw.**

Prima di tutto, cabliamo i nostri componenti elettronici sulla breadboard. È consigliabile spegnere il RaspberryPi, mentre lo fate. Come mostrato nello schema, colleghiamo la gamba più corta (negativo) del LED al pin GPIO 6 del Pi. Colleghiamo la gamba più lunga (positivo) a una resistenza da 333Ω; l' altra estremità di quest'ultima sarà collegata al pin 3V3 del Pi. Questa configurazione è nota come 'attivo basso', secondo la quale, quando il pin GPIO è impostato a livello logico basso, la corrente circolerà e il LED e si accenderà; quando il pin ha livello logico alto, invece, si spegnerà.

Successivamente, andremo a cablare il pulsante, posizionandolo a cavallo della scanalatura al centro della breadboard. Un pin è collegato al GPIO 5, mentre l'altro (sullo stesso lato) è collegato a massa. Di nuovo, questa è una configurazione

Il semplice circuito della breadboard è collegato ai pin GPIO sul Raspberry Pi

attivo-basso: il segnale è alto per impostazione predefinita e diventa basso quando si preme il pulsante. Quando hai terminato i collegamenti, accendi il Raspberry Pi. Nota che il Pi dovrebbe essere in headed mode (che è la modalità predefinita) per questo progetto. Se così non fosse, la dovrai impostare con la PowerShell sul tuo PC principale Windows 10 - vedi **magpi.cc/29K4XWH**.

Ora, su l PC con Windows 10, scarica gli esempi di codice, incluso PushButton per questo progetto, dalla repository GitHub: **magpi.cc/29Qwut8**. Premi il pulsante destro del mouse sul file zip e estrai tutti i file, quindi apri **samples-develop\PushButton\CS\PushButton.csproj** (Il file di progetto di Visual C#) in Visual Studio; se non già fatto, ti verrà chiesto di attivare la modalità sviluppatore sul PC.

Scegli ARM per l'architettura di destinazione. Vai su **Build>Build Solution**. Questo processo può richiedere un po' di tempo. Al termine, seleziona **Macchina Remota** dal menù a tendina Dispositivo, inserisci l'indirizzo IP del tuo Raspberry Pi, e seleziona **Nessuna** per il tipo di autenticazione. Premere F5 per compilare ed eseguire il debug. Una volta che è correttamente compilato, prova a premere il pulsante sul tuo circuito; il LED deve accendersi. Ogni volta che premi il pulsante, il LED dovrebbe cambiare di stato.

Illustrare il codice

Diamo uno sguardo al codice nell'elemento **MainPage ()** per capire come questo lavori. Nella parte superiore, apriamo le risorse del pin GPIO che useremo...

```
  buttonPin = gpio.
OpenPin(BUTTON_PIN);
  ledPin = gpio.OpenPin(LED_
PIN);ode>
```

Assegniamo al pin del LED un livello alto, per assicurarci che sia inizialmente spento. Quando cambiamo il drive mode in output, esso immediatamente sarà cambiato nel livello associato al pin:

```
  ledPin.Write(GpioPinValue.
High);
  ledPin.
SetDriveMode(GpioPinDriveMode.
Output);ode>
```

Dopo questo, abbiamo impostato il pin del pulsante. Dal momento che Raspberry Pi 2 e 3 hanno resistenza di pull-up integrata, che possiamo attivare via software, non abbiamo bisogno di aggiungerne una esterna.

```
  if (buttonPin.IsDriveMode
Supported(GpioPinDriveMode.
InputPullUp))
   buttonPin.
SetDriveMode(GpioPinDriveMode.
InputPullUp)

   buttonPin.
SetDriveMode(GpioPinDriveMode.
Input);ode>
```

Poi, colleghiamo l'interrupt di ascolto del GPIO, che viene richiamato ogni volta che il pin del pulsante cambia di stato. Abbiamo anche impostato il DebounceTimeout a 50ms per filtrare eventi spuri causati da disturbi elettrici.

```
  buttonPin.DebounceTimeout =
TimeSpan.FromMilliseconds(50);
  buttonPin.ValueChanged +=
buttonPin_ValueChanged;ode>
```

Nel gestore dell'interrupt del pulsante, dobbiamo trovare ile rampe del segnale del GPIO per determinare se il pulsante è stato premuto o rilasciato. Se premuto, dobbiamo cambiare lo stato del LED:

```
  private void buttonPin_
ValueChanged(GpioPin sender,
GpioPinValueChangedEventArgs e)
  {
   if (e.Edge == GpioPinEdge.
FallingEdge)
   {
   ledPinValue = (ledPinValue ==
GpioPinValue.Low) ?
   GpioPinValue.High :
GpioPinValue.Low;
   ledPin.Write(ledPinValue);
   }ode>
```

Infine, vogliamo aggiornare l'interfaccia utente con lo stato attuale del pin, quindi invochiamo una operazione di aggiornamento sul thread UI

Questo tutorial è stato adattato da Push Button Sample, dl team Windows IoT, composto da Daniel Kuo, Tycho's Nose e Liz George, dal sito progetti di Microsot: magpi.cc/29K5xne

Linguaggio
>C#

NOME DEL FILE:
PushButton.csproj

DOWNLOAD:
magpi.cc/29Qwut8

Seleziona 'Macchina Remota' e immetti l'indirizzo IP o il nome del tuo Raspberry Pi

Il codice che controlla il circuito può essere trovato nell'elemento MainPage ( ) del progetto

ROB ZWETSLOOT

Riparatore, a volte maker, altre volte cosplayer, e tutto il resto del tempo, capo redattore di ***The MagPi***
magpi.cc

LED CONNESSO A INTERNET

“Connettiti alla **Internet delle cose!**”

Prendi informazioni da internet per accendere un LED, connettiti all'internet delle cose e entra a far parte della Mappa Mondiale dei !! broken!!

Cosa Serve

- 1× breadboard
- 1× LED
- 1× resistenza da 560ΩΩ
- 2× cavallotti

Ora sai come accendere un LED premendo un pulsante su Windows 10 IoT Core. Vediamo ora come utilizzare la parte IoT del software per ottenere un LED controllato da internet !

Prepara il Pi

Spegni il Raspberry Pi e cabla il circuito come mostrato nello schema Fritzing. E' decisamente semplice, quindi non dovrebbe impegnarti per molto tempo.

Successivamente, dovrai scaricare un po' di codice per avete tutto quello che ti serve: **MainPage.cs**, che è una piccola parte del progetto. Prendilo da qui : **magpi.cc/2a5G0rP**. Apri **Lesson_201.sln** e poi apri **MainPage.xaml.cs**. Quest'ultimo deve essere modificato per essere in grado di eseguire lo script LED che vogliamo scrivere, quindi sostituisci tutto il codice con la versione completa da qui: **magpi.cc/2a7rNgL**.

Una volta fatto, possiamo creare il file classe. Dal menu principale , selezionare **Progetto > Aggiungi Classe** ... Si aprirà il pannello Aggiungi Nuovo Elemento, di default come classe Visual C #. Come
accennato prima, la stiamo nominando **InternetLED.cs**.

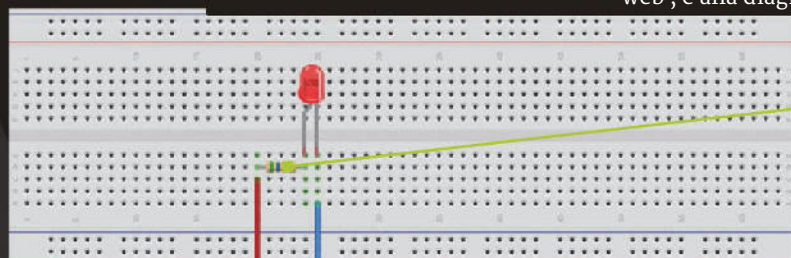
Per cominciare , abbiamo bisogno di aggiungere le righe 'using' all'inizio del file, in modo che il codice può fare riferimento al GPIO del dispositivo , alle interfacce web , e alla diagnostica di sistema.

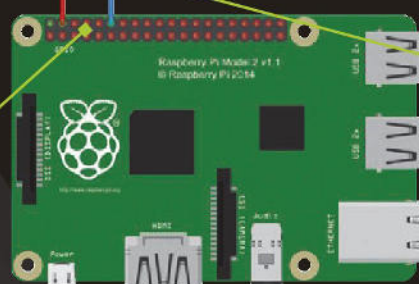
```
using System;
using System.Diagnostics;
using System.Net.Http;
using System.Threading.Tasks;
using Windows.Devices.Gpio;

namespace Lesson_201
{
```

All'interno delle parentesi della classe, aggiungi le seguenti le linee. le prime tre righe servono per controllare il GPIO, la riga successiva ci permette di aggiungere un ritardo , se non otteniamo una risposta per la chiamata API web, e le ultime due righe memorizzare lo stato del LED (acceso o spento).

```
class InternetLed
  {
      private GpioController
gpio;
      private GpioPin
LedControlGPIOPin;
      private int
LedControlPin;
      private const int
DefaultBlinkDelay = 1000;
      public enum eLedState {
Off, On };
      private eLedState _
LedState;
```

Ora, aggiungere il codice costruttore della classe, che conterrà il numero del pin utilizzato per controllare il LED, interagendo con la sezione dedicata al controllo del GPIO, della classe

La resistenza collega il piedino positivo del LED con l'ingresso 5V, che alimenta il LED

I piedino negativo del LED è collegato al GPIO 12 utilizzando un cavallotto, che controlla quando si accende e si spegne

Noi stiamo usando il pin 4, che fornisce 5V, e il pin GPIO programmabile 12, in questa configurazione

Join 4,000 other makers on the World Map of Makers, and light up their LEDs too!

```
  public InternetLed(int
ledControlPin)
    {
      Debug.
WriteLine("InternetLed::New
InternetLed");
      LedControlPin =
ledControlPin;
    }
```

Ora aggiungiamo il codice che inizializza la comunicazione con il GPIO del Pi. Questo, prima imposta il GPIO da utilizzare con il LED, e poi ci permette di vedere il valore corrente del pin:

```
  public void InitalizeLed()
  {
    Debug.WriteLine("InternetLed
::InitalizeLed");
    gpio = GpioController.
GetDefault();
    LedControlGPIOPin = gpio.
OpenPin(LedControlPin);
    LedControlGPIOPin.
SetDriveMode(GpioPinDriveMode.
Output);
    GpioPinValue startingValue =
LedControlGPIOPin.Read();
    _LedState = (startingValue
== GpioPinValue.Low) ?
eLedState.On : eLedState.Off;
  }
```

Questa parte ci permette di interagire con il LED, a seconda del suo stato:

```
  public eLedState LedState
  {
    get { return _LedState; }
    set
    {
      Debug.WriteLine("Interne
tLed::LedState::set " + value.
ToString());
      if (LedControlGPIOPin !=
null)
      {
        GpioPinValue newValue
= (value == eLedState.
On ? GpioPinValue.High :
GpioPinValue.Low);
        LedControlGPIOPin.
Write(newValue);
        _LedState = value;
      }
    }
  }
```

Ora aggiungiamo una parte che ci permette di far lampeggiare il LED tra acceso o spento. Se il LED è acceso, viene spento. Se è spento, viene acceso. Semplice!

```
  public void Blink()
    {
      if (LedState == eLedState.
On)
      {
        LedState = eLedState.
Off;
      }
      else
      {
        LedState = eLedState.On;
      }
    }
```

Ora possiamo davvero iniziare a utilizzare informazioni da Internet. Inizia con la creazione di un sistema per leggere una pagina web e poi eseguirla. Una volta che è stato letta, l'output è ritornato come stringa al canale di debug, così lo puoi vedere mentre si esegue un progetto.

Infine, determiniamo il valore del ritardo e lo ritorniamo, per l'uso nel resto del codice.

```
  const string
WebAPIURL
= "http://
adafruitsample.
azurewebsites.net/
TimeApi";
  public async Task<int>
GetBlinkDelayFromWeb()
  {
    Debug.WriteLine("Interne
tLed::MakeWebApiCall");
    string responseString = "No
response";
    try
    {
      using (HttpClient client =
new HttpClient())
      {
        responseString
= await client.
GetStringAsync(WebAPIURL);
        Debug.WriteLine(String.
Format("Response string: [{0}]",
responseString));
      }
    }
    catch (Exception e)
    {
      Debug.WriteLine(e.
Message);
    }
    int delay;
    if (!int.
TryParse(responseString, out
delay))
    {
      delay = DefaultBlinkDelay;
    }
    return delay;
  }
 }
}
```

Una volta che tutto il codice è inserito, puoi realizzare la build e far eseguire il codice sul tuo Pi. In caso di problemi, il codice implementa già completamente il controllo errori.

Questo tutorial è stato adattato da World Map of Makers dal team IoT di Windows, composto da David Shoemaker, Anthony Ngu, e Aparajita Dutta, dal sito Microsoft Projects:: magpi.cc/2a5GaPD

Linguaggio
>C#

NOME DEL FILE:
InternetLED.cs

DOWNLOAD:
magpi.cc/2a5Fqdp

ROB ZWETSLOOT
Riparatore, a volte maker, altre volte cosplayer, e tutto il resto del tempo, capo redattore di ***The MagPi***.
magpi.cc

SENSORE DI LUCE PARLANTE

Crea un sensore di luce personalizzabile e parlante che ti fa sapere quando l'illuminazione è sufficiente

Cosa Serve

- 1× breadboard
- 1× sensore di luce (fotoresistenza)
- 1× resistenza da 560 Ω
- cavallotto maschio femmina
- altoparlanti o cuffie (opzionali)

Negli ultimi due tutorial, abbiamo imparato ad usare un tasto per accendere un LED, e poi come utilizzare una chiamata API web per accendere un altro LED. Uniamo ora entrambi questi concetti, per costruire un dispositivo che legge i valori di due potenziometri (resistenze variabili) ed un sensore di luce, ed utilizza queste informazioni per informarti a voce del risultato utilizzando una API text-to-speech online. Gli faremo dire "luminoso", oppure no!

Innanzi tutto, a Raspberry Pi spento, effettua i collegamenti come indicato nel diagramma Fritzing, facendo molta attenzione all'orientamento del chip. Fai riferimento alla *J ǫ ò* nella prossima pagina per le note su come orientarlo. Collega poi un altoparlante o delle cuffie per poter sentire il messaggio.

Il codice principale

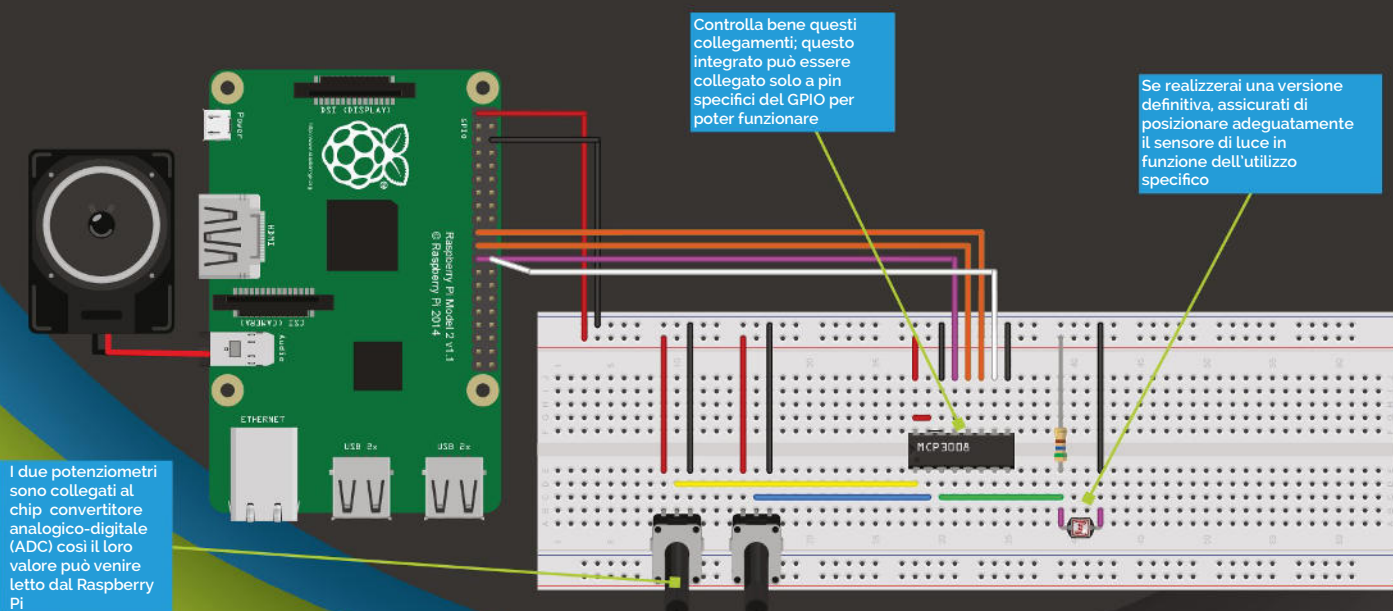
Come nell'ultimo tutorial, scaricheremo il progetto da realizzare. Quindi, prendiamo il tutto ancora da **magpi.cc/2a9OEW1**.

Apri la *P*<<87x ół őh<57* con Visual Studio ed il file *6 &o79&; *hA&6 5h(<*. Dobbiamo però prima fare un po' di modifiche per far funzionare il progetto, e quindi lo facciamo come prima cosa.

Aggiungi le seguenti linee all'inizio della pagina principale, proprio dopo la parentesi graffa {. In primo luogo, dobbiamo leggere la tensione che arriva al chip:

Linguaggio
>C#

NOME DEL FILE:
MainPage.xaml.cs

DOWNLOAD:
magpi.cc/2ac38rr

Credit: lainf

La resistenza variabile, o potenziometro, è un partitore di tensione che, quando routato, permette di avere una tensione variabile

```
  const float ReferenceVoltage =
5.0F;
```

Specifichiamo poi quali sono i canali che devono essere utilizzati. La terza linea è per il sensore di luce :

```
  const byte
LowPotentiometerADCChannel = 0;
  const byte
HighPotentiometerADCChannel =
1;
  const byte CDSADCChannel = 2;
```

Di seguito le stringhe di testo che ci faremo pronunciare dal Pi:

```
  const string
JustRightLightString = "Ah,
just right";
  const string LowLightString =
"I need a light";
  const string HighLightString
= "I need to wear shades";
```

Questa parte aiute a memorizzare lo stato del sensore di luce:

```
  enum eState { unknown,
JustRight, TooBright, TooDark};
  eState CurrentState = eState.
unknown;
```

Istruiamo ora il codice su quale tipo di ADC stiamo utilizzando:

```
  MCP3008 mcp3008 = new
MCP3008(ReferenceVoltage);
```

Dobbiamo ora aggiungere la parte di speech-to-text di Windows 10 Cognitive services, quindi aggiungiamo:

```
  private SpeechSynthesizer
synthesizer;
```

E per finire aggiungiamo del codice che consente di leggere l'ADC ad intervalli di tempo definiti:

```
  public Timer timer;
```

Completata la parte iniziale, di definizione delle classi, andiamo a modificare la pagina principale (public MainPage()), subito dopo la scritta: InitializeComponent();. Quì aggiungiamo una riga per inizializzare il sintetizzatore vocale ed un'altra per inizializzare il convertitore ADC:

```
  synthesizer = new
SpeechSynthesizer();
  mcp3008.Initialize();
```

Settiamo ora il timer col metodo OnNavigateTo. Dopo Debug.WriteLine("MainPage::OnNavigatedTo");, aggiungere quanto segue:

"Un Raspberry Pi che parla!"

```
  timer = new
Timer(timerCallback, this, 0,
1000);
```

Ora abbiamo la richiamata del timer che dobbiamo completare. Dopo la parentesi graffa } seguita da 'return;', aggiungere le seguenti righe di codice. Settiamo l'impostaziuone iniziale del grado di illuminazione:

```
  eState newState = eState.
JustRight;
```

Poi vogliamo leggere i valori che arrivano dall'ADC, quindi scriviamo:

```
  int lowPotReadVal = mcp3008.
ReadADC(LowPotentiometerADCCha
nnel);
  int highPotReadVal = mcp3008.
ReadADC(HighPotentiometerADCCh
annel);
  int cdsReadVal = mcp3008.
ReadADC(CDSADCChannel);
```

Convertiamo queste letture in tensioni per renderne più semplice l' utilizzo:

Alla fine dei cablaggi il tutto potrà sembrare molto aggrovigliato, ma poi utilizzare dei cavetti colorati così da individuare dal colore a quale parte del sistema appartengono.

Fig 1 La tacca sul Chip indica che quel lato è la 'fronte' del chip. Nello schema Fritzing, questa tacca è sul lato sinistro del chip

UTILIZZI PRATICI

Come potrebbe essere usato questo progetto nella vita di tutti i giorni? Beh, ad esempio per essere sicuri che le piante vengano esposte correttamente alla luce.

Oppure si può posizionare il sensore vicino al la TV per capire quando è ora di tirare le tendine per evitare i fastidiosi riflessi!

```
  float lowPotVoltage = mcp3008.
ADCToVoltage(lowPotReadVal);
  float highPotVoltage = mcp3008.
ADCToVoltage(highPotReadVal);
  float cdsVoltage = mcp3008.
ADCToVoltage(cdsReadVal);
```

Poi dobbiamo assicurarci che si riescano a leggere i valori quando il programma è in esecuzione dal debug:

```
  Debug.WriteLine(String.
Format("Read values {0}, {1},
{2} ",
    lowPotReadVal,
highPotReadVal, cdsReadVal));
  Debug.WriteLine(String.
Format("Voltages {0}, {1}, {2}
",
    lowPotVoltage,
highPotVoltage, cdsVoltage));
```

Possiamo ora verificare se l'intensità della luce è inferiore ai valori impostati tramite i potenziometri:

```
  if (cdsVoltage < lowPotVoltage)
  {
    newState = eState.TooDark;
  }
```

Il prossimo passo è verificare se il valore è troppo alto:

```
  if (cdsVoltage >
highPotVoltage)
  {
    newState = eState.TooBright;
  }
```

E poi dobbiamo fare il modo che il codice sappia di dover utilizzare un metodo diverso per decidere cosa fare con lo stato della luce:

```
  await
CheckForStateChange(newState);
```

Il codice CheckForStateChange è ora quasi completo in questo esempio, ma è necessario aggiungere quanto segue sotto 'use another method to wrap the speech synthesis functionality':

```
  await TextToSpeech(whatToSay);
```

Aggiungiamo ora la parte che permette al Pi di parlarci! Sotto async, dove dice 'insert code', Inserire il codice seguente, che in primo luogo converte il testo in una stringa, e poi riproduce l'audio:

```
  SpeechSynthesisStream
synthesisStream;
  synthesisStream = await
synthesizer.SynthesizeTextToStr
eamAsync(textToSpeak);

  media.AutoPlay = true;
  media.
SetSource(synthesisStream,
synthesisStream.ContentType);
  media.Play();
```

Programmare l'ADC

Ora che abbiamo ultimato il codice della MainPage, passiamo al codice per la parte ADC. Apri ***Q GTôł ł ûł(*** < dal file di progetto.

Partiamo creando un metodo di inizializzazione per settare la comunicazione con il bus SPI Su cui è connesso il chip ADC. Subito dopo la riga 'Debug.WriteLin e ("MCP3008::Initialize");' aggiungere quanto segue, che inizializza la configurazione del bus SPI bus e specifica la velocità del clock dell'ADC:

```
  try
  {
    var settings = new
SpiConnectionSettings(SPI_CHIP_
SELECT_LINE);
    settings.ClockFrequency =
3600000;
    settings.Mode = SpiMode.
Mode0;
```

Questa riga elenca tutti dispositivi SPI nel sistema:

```
    string aqs = SpiDevice.
GetDeviceSelector();
```

Questa riga individua il controller del bus SPI dalla riga precedente:

Linguaggio

>C#

NOME DEL FILE:
MCP3008.cs

DOWNLOAD:
magpi.cc/2agTxOR

Ci sono un sacco di tutorial su GitHub dove si trova questo progetto. Dagli una occhiata, se vuoi approfondire!

```
    var dis = await
DeviceInformation.
FindAllAsync(aqs);
```

Il resto del codice crea un dispositivo SPI utilizzando tutte le informazioni appena ottenute:

```
    mcp3008 = await SpiDevice.
FromIdAsync(dis[0].Id, settings);

    if (mcp3008 == null)
    {
      Debug.WriteLine(
     "SPI Controller {0} is
currently in use by another
application. Please ensure that
no other applications are using
SPI.",
      dis[0].Id);
      return;
    }

  }
  catch (Exception e)
  {
    Debug.WriteLine("Exception:
" + e.Message + "\n" +
e.StackTrace);
    throw;
  }
```

Adesso abbiamo le informazioni riguardo al chip, e siamo in grado di aggiungere il codice per poterlo leggere:

```
  public int ReadADC(byte
whichChannel)
  {
    byte command =
whichChannel;
    command |= MCP3008_
SingleEnded;
    command <<= 4;

    byte[] commandBuf = new
byte[] { 0x01, command, 0x00 };

    byte[] readBuf = new byte[]
{ 0x00, 0x00, 0x00 };

    mcp3008.
TransferFullDuplex(commandBuf,
readBuf);

    int sample = readBuf[2] +
((readBuf[1] & 0x03) << 8);
    int s2 = sample & 0x3FF;
    Debug.Assert(sample == s2);
    return sample;
  }
```

Vi ricordate quando abbiamo convertito il valore di ingesso in una tensione per facilitarne la lettura? Dobbiamo aggiungere un metodo che ci aiuti in questo:

```
  public float ADCToVoltage(int
adc)
  {
    return (float)adc *
ReferenceVoltage / (float)Max;
  }
```

E' tempo di giocare

Il tuo progetto è ultimato! Avvia il programma in un ambiente con una illuminazione normale. La videata ti mostrerà le tensioni lette dall' ADC, i valori inferiore e superiore delle resistenze e la luce del sensore. Dovrebbe essere simile a quanto riportato di seguito:

```
  MainPage::timerCallback
  Read values 211, 324, 179
  Voltages 1.031281, 1.583578,
0.8748778
```

Ruota il potenziometro con il valore di resistenza più basso, ed osserva come cambia il primo numero. Regolalo in modo tale che sia appena un poco inferiore alla lettura del sensore. Regola ora nello stesso modo l'altro potenziometro ma in modo che il valore sia un poco maggiore di quello del sensore. Questa è ora il range in cui la luce è 'giusta'.

Ora, coprendo il sensore con una mano, vedrai comparire nella videata 'Ho bisogno di luce'; sentirai anche lo stesso messaggio in cuffia o sugli altoparlanti se sono collegati. Sperimenta coprendo e illuminando il sensore per verificarne la sensibilità.

Questo tutorial è stato adattato da Bright or Not? dal team Windows IoT costituito da David Shoemaker, Anthony Ngu, e Aparajita Dutta, dal sito Microsoft Projects : magpi.cc/2abUvx0

Il sensore di luce, anche detto fotoresistenza, cambia la sua resistenza in funzione alla quantità di luce che lo investe. Al pari di una resistenza variabile, cambia la tensione in uscita.